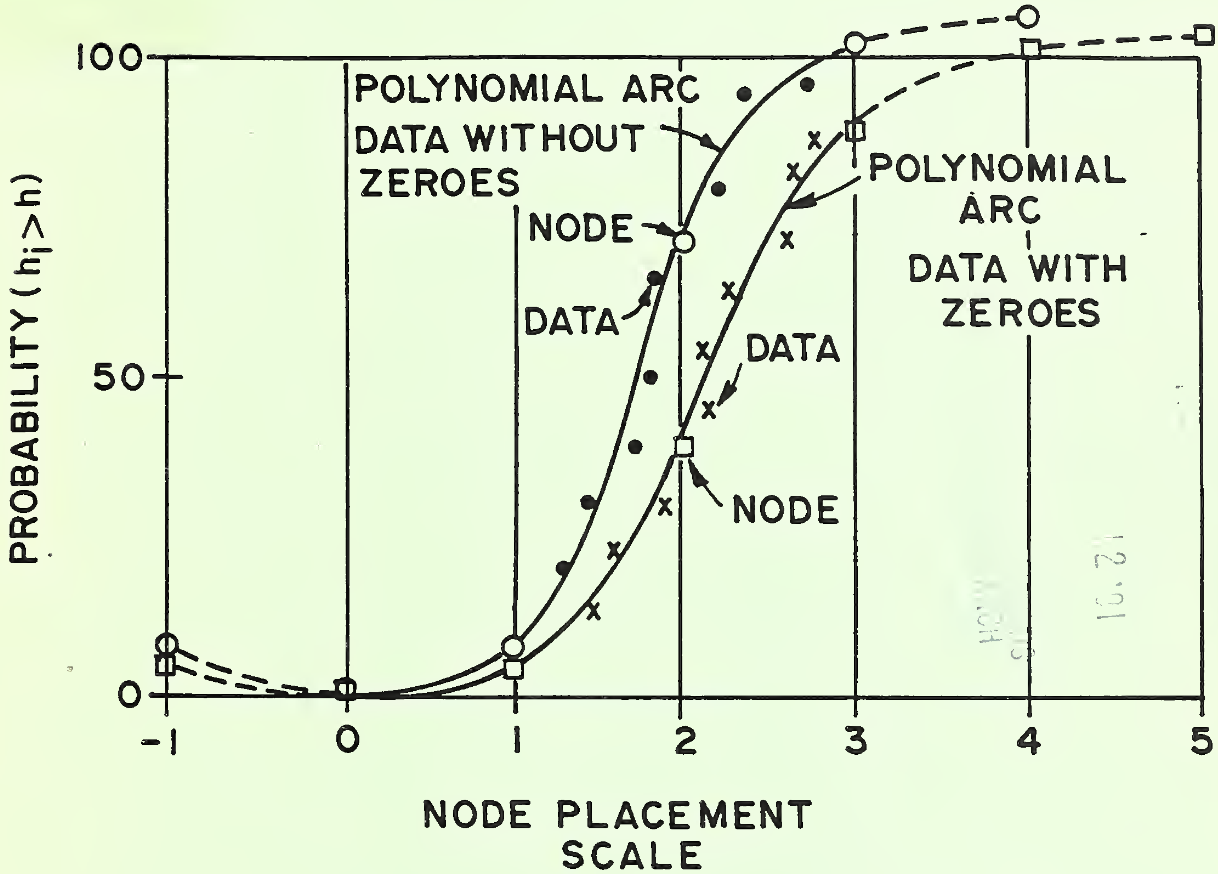
COMPUTER PROGRAMS FOR ANALYSIS AND SIMULATION OF PROBABILITY DISTRIBUTIONS USING SLIDING POLYNOMIALS



Southern Piedmont Conservation Research Center
Agricultural Research Service, USDA
Watkinsville, Georgia 30677

RESEARCH REPORT
No. IRC 070184
July 1984

Research Report[1/]
No. IRC 070184

Computer Programs for Analysis and Simulation of Probability Distributions Using Sliding Polynomials

A. W. Thomas
Agricultural Engineer
Southern Piedmont Conservation Research Center
Agricultural Research Service, USDA
Watkinsville, GA 30677

and

W. M. Snyder
Hydrologic Consultant
275 Gatewood Circle
Athens, GA 30606

July 1984

[1/] This report is intended to provide computer programs for probabilistic analysis and synthesis of experimental data that were derived from papers by Snyder and Thomas, 1983; and Thomas and Snyder, 1984a; 1984b.

INTRODUCTION

A series of three recent papers (Snyder and Thomas, 1983; Thomas and Snyder, 1984a; 1984b) has introduced methodology for performing probabilistic analysis and synthesis of experimental data without assuming that a particular distribution function is appropriate to a particular data set. The new method is mathematically form-free. Statistical samples are smoothed by the piece-wise sliding polynomials, yielding form-free probability distributions. These probability curves can be used for essentially all of the applications purposes of conventional distributions, such as normal, log-normal, or gamma functions. The methods are computer oriented. Therefore, this research report has been prepared for the convenience of potential users. General background material on sliding polynomials may be found in earlier papers by Snyder, 1976 and Snyder, 1980.

Discussion of the smoothing methodology will be brief, intended only to provide understanding sufficient for use and adaptation of programs. The user is referred to the references for fuller understanding.

Smoothing of statistical samples with sliding polynomials requires a mathematical transform of the scale of the original data, called variate h, to a scaled abstract variate called v. This transformation was developed so that the smoothing can incorporate boundary controls at infinite values of h. This transformation is gradually being standardized through additional research. The programs covered in this report are based on the standardized transform.

METHOD OF ANALYSIS

The method of smoothing statistical samples with sliding polynomials and with incorporation of boundary controls is shown geometrically in Figure 1. Most real hydrologic data can vary between zero and + ∞. An exception is temperature, where real values may be negative. With the usual zero discontinuity, the 0 to + ∞ scale for the hydrologic variable, h, is shown at the bottom right of Figure 1. h is first converted to a standardized variate, h' as in equation 1.

$$h' = \frac{h - \overline{h}}{ks} \qquad (1)$$

$\overline{h}$ is the mean of h, s is the standard deviation of h, and k is an empirical parameter. A value of 2 for k works for many data sets.

h' is converted to v by equation 2.

$$\begin{aligned} v &= 4.0 - 2.5 \exp (0.91629h') && h \leq \overline{h} \\ & && h' \leq 0 \\ & && v \geq 1.5 \qquad (2) \\ v &= 1.5 \exp (-1.52715h') && h \geq \overline{h} \\ & && h' \geq 0 \\ & && v \leq 1.5 \end{aligned}$$

Equation 2 expresses a compound curve composed of two descending exponentials common and tangent at h' = 0, v = 1.5. For calibration, the curve is required to pass through the point h' = -1.0, v = 3.0.

In the upper section of Figure 1, it can be seen that three arcs, or spans, of the sliding polynomials are made to cover the possible range of h from zero to + ∞. When h is zero, v is given by equation 3.

$$v(h = 0) = 4 - 2.5 \exp (-0.91629\ \overline{h}/ks) \qquad (3)$$

v (h = 0) will vary from sample to sample as $\bar{h}$ and s vary. The interval from v = 0 to v (h = 0) is divided into three equal segments to place nodes at uniform interval in v-scale. Nodes 0, 1, 2, and 3 are so located. Using the same interval, boundary nodes are placed at -1, 4, and 5. The location of these boundary nodes has no meaning in v-scale, or h-scale. They are necessary to shape the sliding polynomial arcs.

Class limits for the sample are defined by dividing each of the three polynomial spans, 0-1, 1-2, and 2-3, into tenths, yielding 30 classes of uniform width in v-scale. These class limits then are converted from v-values back to h-values with equations 1 and 2.

The statistical sample is tallied into the 30 classes. The classes are subtotalled, starting from large h back toward, but not including, zero. The subtotals, divided by the sample size, and multiplied by 100, gives a sample probability of any individual h being greater than the smaller class limit in h. The sample is then ready for smoothing by least squares.

Sliding polynomial smoothing provides values of the nodes at 1, 2, 3, and 4 on the node placement scale. The nodal value at zero placement must be zero, since the smoothing curve must rise from zero when h is + ∞. An additional boundary condition, that the slope of the smoothing curve must be zero when h is + ∞, is imposed on the smoothing by requiring the node at -1 to equal the node at +1 placement.

A modification to the smoothing process must be made if the sample contains zero values of h. In the upper part of Figure 1, it will be noted that the node at 3 may be greater than or less than 100 percent probability. If this node is greater than or equal to 100, then the sample does not contain zeroes. The 30 classes will contain all the

values in the sample and will reach 100 percent. The point where the smoothing curve crosses the 100 percent line can be reflected back to h and is an estimate of the lowest value of h.

If the sample contains zeroes and the zeroes are excluded from the 30-class tally, the sample probabilities will not reach 100 percent, and node three will be less than 100. In this case, the zeroes are considered to compose a 31st class, and provide a sample point on the 100 percent probability line. This requires an extra node placed at 5, and also requires an iterative least squares smoothing to find where a fictitious arc crosses the 100 percent line. This arc is called fictitious because it implies values of h less than zero, and this is impossible. The method has the advantage, however, of providing analytical integrity across the non-zero and zero items in a mixed sample.

Program 1, below, performs smoothing of data without zeroes. Program 2 performs smoothing of data with zeroes. Note that no distinction is made between extreme-value maxima, extreme-value minima, nor total samples.

Further work may be needed on the best value of the empirical parameter k. It could possibly be a function of the third moment of the sample. The transform from h to v may also need some relatively minor modifications. In particular the common point of the two exponential limbs may need to be moved from v=1.5 toward v=2.0 for some highly skewed distributions of maxima. Also, the calibration point could be changed from h'=-1.0, v=3.0. The only objective of the h to v transform is to produce a fairly smooth sample probability rising from zero.

METHOD OF SIMULATION

Programs 1 and 2 analyze one historical sample. The derived smoothing curves are estimates of population probabilities as provided by the sample. We often need estimates of possible variability if such samples were to be repeated. For example, we might be interested in possible future daily or monthly rainfall variability. We might be interested in how many times a critical value could be equalled or exceeded in, say, 100 such future samples. These values form confidence intervals and tolerance limits on values for planning and design.

The smoothing curves derived from samples by Program 1 and Program 2 can be used to synthesize possible future samples. Random numbers can be generated in a computer program. Each such random number, in a scale from zero to 100, may be considered a probability. Each value of probability has an equal chance of occurrence. These random probabilities can be reflected through the smoothing curves by reverse interpolation to yield values of v. These random v-values can then be transformed to random h's. As an example, if h represented the annual maximum flood, then 50 random h's could be called a simulated 50-year record of annual floods.

Program 3, below, performs simulation of samples with no zeroes. The smoothing curve would have been derived using Program 1 in the analysis of an historical sample.

Program 4 performs simulation of samples with zeroes. Program 2 would have been used for analysis.

Both programs allow for simulation of a varying number of simulated samples of a designated number of items each. Again, an example, if h represented the annual maximum flood, 100 sets of 50 random h's could

be called 100 possible future 50-year simulated flood records. Each of the 100 are placed in order of magnitude. For rank number 1 of the 50, one has thus a 100 item distribution of the 1-in-50 year flood. The 1-in-50 sets a risk level. The 100 item distribution is the uncertainty associated with the designated risk.

REFERENCES

1. Snyder, W. M. and A. W. Thomas. 1983. Return Period Analysis with Sliding Polynomials. Trans. ASAE. 26(6):1732-1737.

2. A. W. Thomas and W. M. Snyder. 1984a. Return Period Analysis of Minimum Events Using Sliding Polynomials. Trans. ASAE. 27(2): 464-469.

3. A. W. Thomas and W. M. Snyder. 1984b. Testing the Representativeness of Short Period Records through Simulation. Trans. ASAE. 27(4):Unassigned.

BACKGROUND MATERIALS

1. Snyder, Willard M. 1976. Interpolation and Smoothing of Experimental Data with Sliding Polynomials. USDA, Agricultural Research Service, ARS-S-83. 34 p.

2. Snyder, Willard M. 1980. Smoothed Data and Gradients Using Sliding Polynomials with Optional Controls. Water Resourc. Bull. 16(1):22-30.

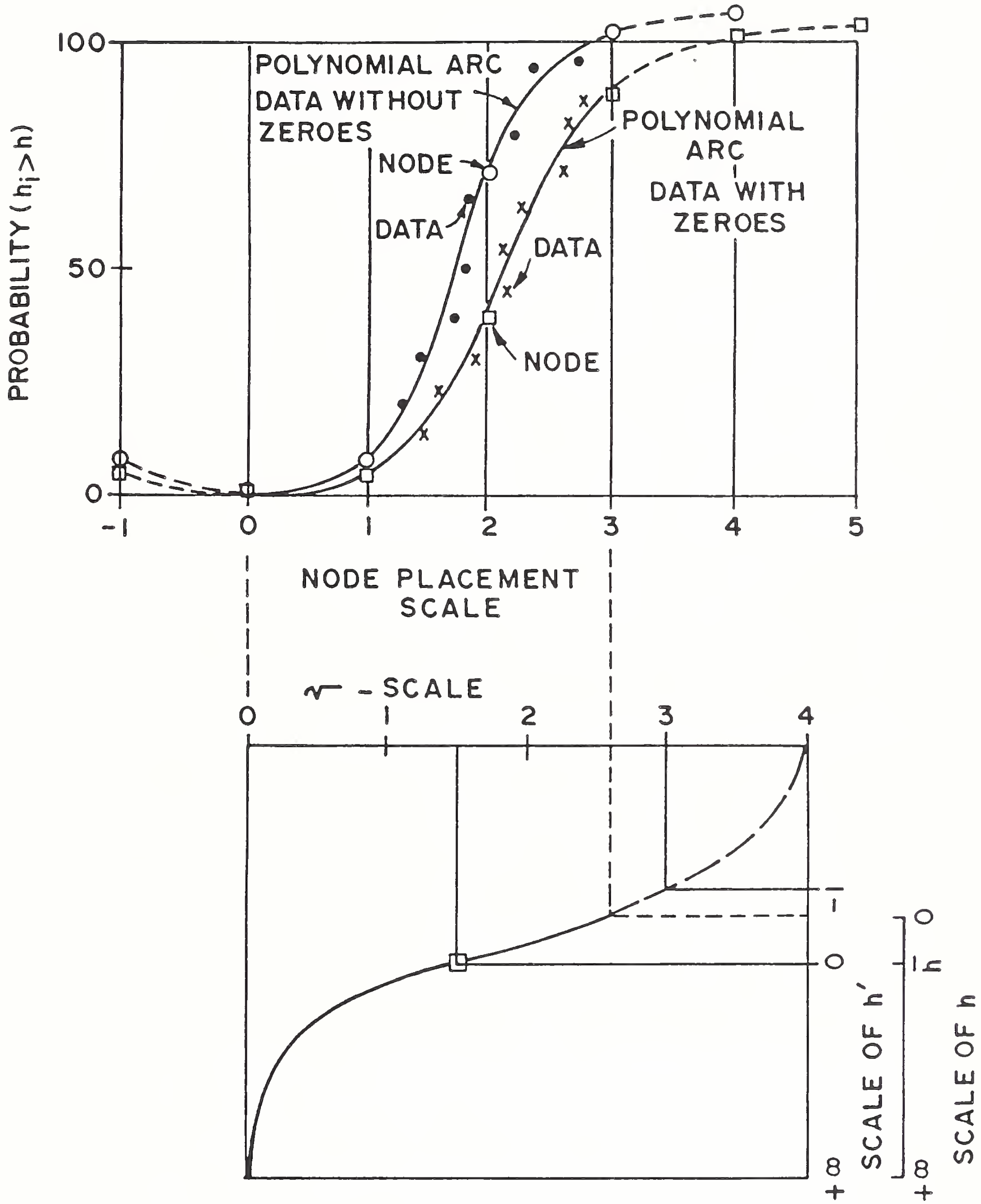


Figure 1. Geometrical Schematic of h to v Transform for Placement of Nodes.

APPENDIX

The Appendix includes four program listings and, notes and sample outputs of each.[1/]

[1/] These programs are presented only for the convenience of potential users. While the programs have been run and tested on various data sets the originators of the programs assume no responsibility for either their accuracy or adequacy. Such responsibility must rest solely on the user. We stand ready to assist and advise within the restrictions imposed by our operating resources. Programs 1 and 2 are written in CYBER BASIC. Programs 3 and 4 are written in FORTRAN G (Trade name is included for the benefit of the reader and does not imply an endorsement or preferential treatment of the named products).

PROGRAM 1

To be used on data sets with few zeroes. Node No. 3 must be $\geq$ 100 probability. (See Figure 1).

This program finds four nodes of the sliding polynomials with data organized into 30 classes.

Boundary condition is set so that the sliding polynomial smoothing curve is zero with zero slope at $+\infty$ of the data scale.

Use Program 3 for simulation with the nodes derived from this fitting.

```
00010 REM  ANALYSIS WITH FEW ZEROES
00020  REM READ FROM FILE #1  AND WRITE TO FILE #5
00030 OPTION BASE 1
00040 FILE #1="PER1"
00050 RESTORE #1
00060 FILE #5="SMDATA"
00070 RESTORE #5
00080 DIM H(200),W(4,4),M(35),C(35),P(35),Y(6)
00090 DIM R(35),X(35),T(35),V(35),S(4)
00100 DIM U(35)
00110 REM  INITIALIZE ARRAYS
00120 FOR I=1 TO 30
00130 NODATA 00290
00140 READ R(I),X(I),T(I),V(I)
00150 NEXT I
00160 DATA 0.028,-0.0045,0,0,0.104,-0.016,0,0,0.216,-0.0315,0,0
00170 DATA 0.352,-0.048,0,0,0.5,-0.0625,0,0,0.648,-0.072,0,0
00180 DATA 0.784,-0.0735,0,0,0.896,-0.064,0,0,0.972,-0.0405,0,0
00190 DATA 1,0,0,0,0.9765,0.0685,-0.0045,0,0.912,0.168,-0.016,0
00200 DATA 0.8155,0.2895,-0.0315,0,0.696,0.424,-0.048,0,0.5625
00210 DATA 0.5625,-0.0625,0,0.424,0.696,-0.072
00220 DATA 0,0.2895,0.8155,-0.0735,0,0.168,0.912,-0.064,0,0.0685
00230 DATA 0.9765,-0.0405,0,0,1,0,0,-0.0405,0.9765,0.0685,-0.0045
00240 DATA -0.064,0.912,0.168,-0.016,-0.0735,0.8155,0.2895,-0.0315
00250 DATA -0.072,0.696,0.424,-0.048,-0.0625,0.5625,0.5625
00260 DATA -0.0625,-0.048,0.424,0.696,-0.072,-0.0315,0.2895,0.8155
```

```
00270 DATA -0.0735,-0.016,0.168,0.912,-0.064,-0.0045,0.0685,0.9765
00280 DATA -0.0405,0,0,1,0
00290 FOR I=1 TO 4
00300 FOR J=1 TO 4
00310 READ W(I,J)
00320 NEXT J
00330 NEXT I
00340 DATA 0.133799,-0.0277062,0.00404919,-0.261391
00350 DATA -0.0277062,0.157706,0.0568727,1.86251
00360 DATA 0.00404919,0.0568727,0.746445,9.06661
00370 DATA -0.261391,1.86251,9.06661,161.165
00380 PRINT #5,, "DEFICIENT RAIN PERIOD #1"
00390 :###   ###.###        ###        ###.#####        ###.####
00400 :#####
00410 PRINT #5,
00420 PRINT #5, "INPUT HYDROLOGIC SAMPLE"
00430 PRINT #5
00440 REM  INPUT N NUMBER OF DATA
00450 INPUT #1,N
00460 PRINT #5, "NUMBER OF ITEMS IN SAMPLE.";N
00470 PRINT #5
00480 REM  PARAMETER K=2.0
00490 INPUT #1,K
00500 REM INPUT DATA INTO H( )
00510 FOR I=1 TO N
00520 INPUT #1,H(I)
```

```
00530 PRINT #5 USING 00400, H(I);
00540 NEXT I
00550 PRINT #5
00560 S1=0
00570 S3=0
00580 REM S1=SUM OF DATA, S3=SUM OF SQUARES
00590 FOR I=1 TO N
00600 S1=S1+H(I)
00610 S3=S3+H(I)*H(I)
00620 NEXT I
00630 H1=S1/N
00640 S5=SQR((S3-S1*S1/N)/(N-1))
00650 PRINT #5
00660 PRINT #5, " SAMPLE AVERAGE IS";H1
00670 PRINT #5
00680 PRINT #5, " SAMPLE SD IS ";S5
00690 I=0
00700 S4=K*S5
00710 V0=4-2.5*EXP(-0.91629*H1/S4)
00720 V0=INT(100*V0+0.5)/100
00730 V2=V0/30
00740 FOR V1=V2 TO V0 STEP V2
00750 I=I+1
00760 IF V1>1.5 THEN 00800
00770 REM M( )=CLASS LIMITS
00780 M(I)=LOG(V1/1.5)*S4/(-1.52715)+H1
```

```
00790 GOTO 00810
00800 M(I)=LOG((4-V1)/2.5)*S4/0.91629+H1
00810 NEXT V1
00820 FOR L=1 TO 35
00830 C(L)=0
00840 NEXT L
00850 REM COMPUTE CLASS FREQUENCY, C( )
00860 FOR I=1 TO N
00870 K=1
00880 IF H(I)<M(K) THEN 00910
00890 C(K)=C(K)+1
00900 GOTO 00930
00910 K=K+1
00920 GOTO 00880
00930 NEXT I
00940 FOR L=1 TO 30
00950 REM  U( )=CLASS SAMPLE
00960 U(L)=C(L)
00970 NEXT L
00980 A1=0
00990 REM COMPUTE CLASS PROBABILITY, P( )
01000 FOR I=1 TO 30
01010 A1=A1+C(I)
01020 P(I)=A1/N
01030 NEXT I
01040 FOR I=1 TO 4
```

```
01050 S(I)=0
01060 NEXT I
01070 FOR I=1 TO 30
01080 S(1)=S(1)+R(I)*P(I)
01090 S(2)=S(2)+X(I)*P(I)
01100 S(3)=S(3)+T(I)*P(I)
01110 S(4)=S(4)+V(I)*P(I)
01120 NEXT I
01130 FOR I=1 TO 4
01140 Y(I)=0
01150 FOR J=1 TO 4
01160 Y(I)=Y(I)+W(I,J)*S(J)
01170 NEXT J
01180 NEXT I
01190 PRINT #5
01200 PRINT #5, " SLIDING POLYNOMIAL ORDINATES"
01210 FOR L=1 TO 4
01220 PRINT #5, Y(L)
01230 NEXT L
01240 FOR L=1 TO 4
01250 K=5-L
01260 Y(K+2)=Y(K)
01270 PRINT #5
01280 NEXT L
01290 Y(2)=0
01300 Y(1)=Y(3)
```

```
01310 REM COMPUTE SMOOTH PROBABILITY, C( )
01320 FOR K=2 TO 4
01330 Z=-0.5
01340 A=(9*(Y(K)+Y(K+1))-Y(K-1)-Y(K+2))/16
01350 B=(11*(Y(K+1)-Y(K))+Y(K-1)-Y(K+2))/8
01360 Q=(Y(K-1)-Y(K)-Y(K+1)+Y(K+2))/4
01370 D=(3*(Y(K)-Y(K+1))-Y(K-1)+Y(K+2))/2
01380 FOR J=1 TO 10
01390 Z=Z+0.1
01400 I1=(K-2)*10+J
01410 C(I1)=((D*Z+Q)*Z+B)*Z+A
01420 NEXT J
01430 NEXT K
01440 PRINT #5,"      CLASS     CLASS     SAMPLE          SMOOTH"
01450 PRINT #5,"     LIMITS     SAMPLE   PROBABILITY    PROBABILITY"
01460 PRINT #5
01470 PRINT #5,"    --------   --------  ------------   --------------"
01480 FOR I=1 TO 30
01490 PRINT #5 USING 00390,I,M(I),U(I),P(I),C(I)
01500 NEXT I
01510 END
```

Notes for Program 1

Line #	Comment
3	Begins arrays at one.
16 - 28	Sliding Polynomial Coefficients for data set organized into 30 classes.
34 - 38	Inverse of Sums-of-Products matrix of Sliding Polynomial Coefficients. This matrix is fixed for 30 classes and 4 nodes.
45 - 54	Input Sample.
56 - 68	Compute average and standard deviation.
69 - 73	Set width of 30 uniform classes in v-scale.
74 - 81	Compute class boundaries in original data scale.
86 - 93	Tally the sample into the 30 classes.
100-103	Accumulate across classes and take ratios.
107-112	Compute the Σ XY vector of least squares.
113-118	Multiply Σ XY vector by inverse matrix to get 4 nodes.
121-130	Re-position the nodes and put in the boundary nodes.
132-143	Lay in the ensemble of 3 Sliding Polynomial arcs across 4 nodes.

Sample Output For Program 1

Deficient Rain Period #1

Input Hydrologic Sample

Number of Items in Sample 116

21	25	5	7	6	44	11	10	6	36	19
8	1	9	8	12	20	3	3	12	17	39
5	2	13	11	15	6	32	30	28	2	1
13	21	27	5	12	42	15	7	31	29	11
36	22	5	18	25	31	3	4	1	2	10
8	14	1	9	18	5	1	2	6	28	11
9	2	5	28	3	1	21	12	7	13	9
2	24	1	11	23	27	13	43	25	9	23
9	7	5	36	7	29	8	1	3	7	16
6	12	6	28	8	28	35	30	12	19	36
2	3	32	24	5	50					

Sample Average is 14.7845

Sample SD is 11.8958

Sliding Polynomial Ordinates
0.259249
0.442444
1.00052
1.86506

	Class Limits	Class Sample	Sample Probability	Smooth Probability
1	59.263	0	0.00000	0.0053
2	48.464	1	0.00862	0.0199
3	42.147	2	0.02586	0.0421
4	37.666	2	0.04310	0.0700
5	34.189	5	0.08621	0.1020
6	31.349	2	0.10345	0.1361
7	28.947	6	0.15517	0.1707
8	26.867	7	0.21552	0.2040
9	25.032	0	0.21552	0.2341
10	23.391	5	0.25862	0.2592
11	21.906	3	0.28448	0.2790
12	20.550	3	0.31034	0.2948
13	19.303	1	0.31897	0.3080
14	18.149	2	0.33621	0.3200
15	17.074	2	0.35345	0.3322
16	16.068	1	0.36207	0.3458
17	15.124	1	0.37069	0.3623
18	14.217	2	0.38793	0.3830

Sample Output For Program 1 (continued)

	Class Limits	Class Sample	Sample Probability	Smooth Probability
19	13.284	1	0.39655	0.4093
20	12.317	4	0.43103	0.4424
21	11.311	6	0.48276	0.4817
22	10.266	5	0.52586	0.5252
23	9.176	2	0.54310	0.5727
24	8.039	6	0.59483	0.6240
25	6.849	11	0.68966	0.6789
26	5.602	6	0.74138	0.7372
27	4.293	8	0.81034	0.7988
28	2.914	7	0.87069	0.8633
29	1.457	7	0.93103	0.9306
30	-0.086	8	1.00000	1.0005

Note: Figure 2 is a plot of this sample output

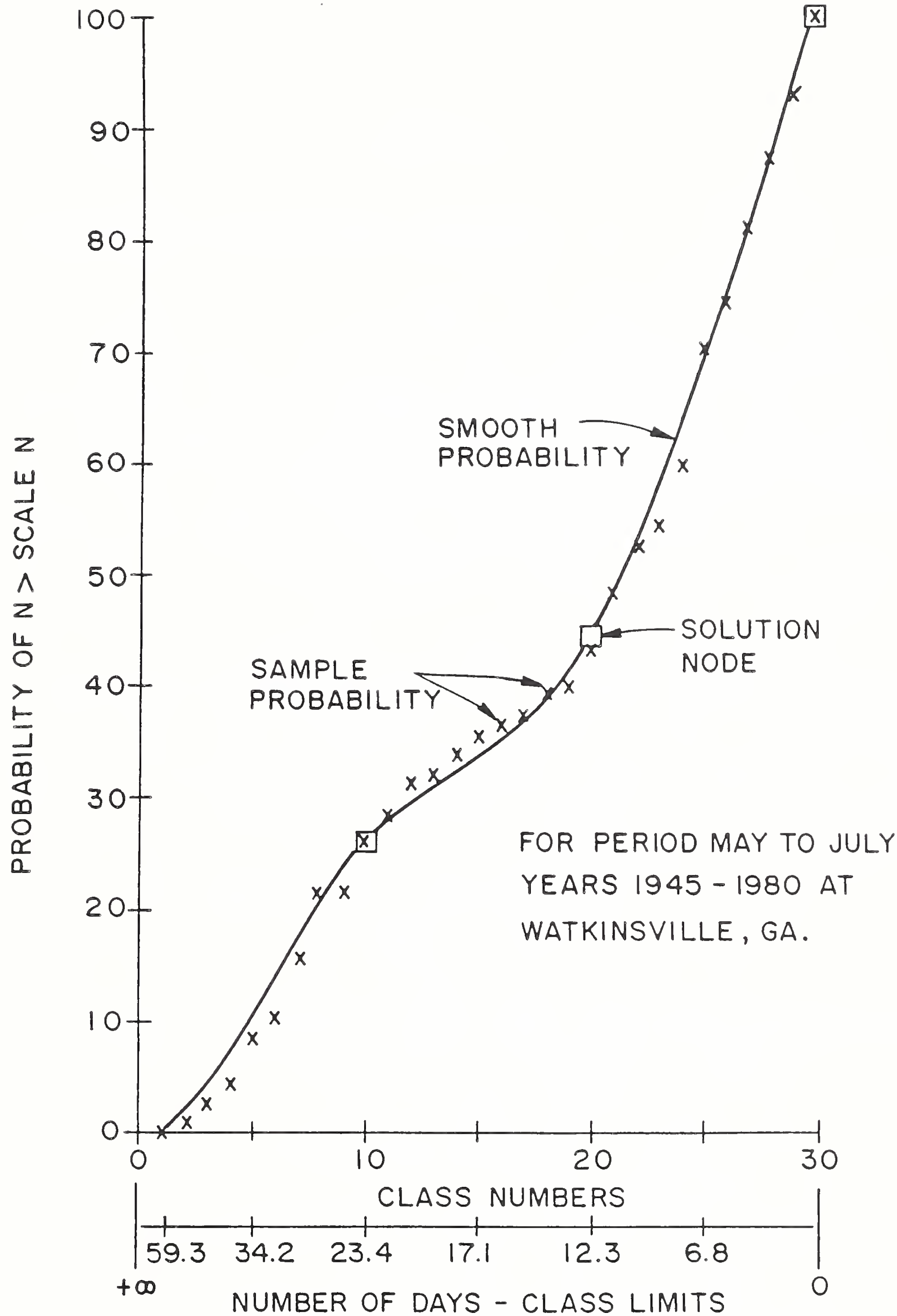


Figure 2. Distribution of Number of Sequential Days with Rain Less Than 1.0 cm.

PROGRAM 2

To be used on data sets containing zeroes.

This program finds five nodes of the sliding polynomials. Non-zero data are organized into 30 classes. Zero data values are put into a 31st class. This program has performed satisfactorily for the authors with data set containing as much as 30 percent zeroes.

Left boundary condition is set so that the sliding polynomial smoothing curve is zero with zero slope at + ∞ of the data scale. Right boundary is free. Difference between 100 and node number 4 is estimate of population proportion of zeroes. A fifth node is necessary and is assumed to lie on a parabola through nodes 2, 3, and 4.

Use Program 4 for simulation with the nodes derived from this fitting.

```
00010  REM  ANALYSIS WITH ZEROES
00020  REM   READ FROM FILE #1   AND WRITE INTO FILE #5
00030 OPTION BASE 1
00040 FILE #1="JUL"
00050 RESTORE #1
00060 FILE #5="SMDATA"
00070 RESTORE #5
00080 DIM H(2000),M(35),C(35),P(4,31)
00090 DIM E(4,4),G(4,4),Q(4,4),Y(7)
00100 DIM A$(80),R(4),D(35),T(40)
00110 FOR I=1 TO 30
00120 FOR J=1 TO 4
00130 NODATA 00300
00140 READ P(J,I)
00150 NEXT J
00160 NEXT I
00170 DATA 0.028,-0.0045,0,0,0.104,-0.016,0,0,0.216,-0.0315,0,0
00180 DATA 0.352,-0.048,0,0,0.5,-0.0625,0,0,0.648,-0.072,0,0
00190 DATA 0.784,-0.0735,0,0,0.896,-0.064,0,0,0.972,-0.0405,0,0
00200 DATA 1,0,0,0,0.9765,0.0685,-0.0045,0,0.912,0.168,-0.016,0
00210 DATA 0.8155,0.2895,-0.0315,0,0.696,0.424,-0.048,0,0.5625
00220 DATA 0.5625,-0.0625,0,0.424,0.696,-0.072
00230 DATA 0,0.2895,0.8155,-0.0735,0,0.168,0.912,-0.064,0,0.0685
00240 DATA 0.9765,-0.0405,0,0,1,0,0,-0.0405,0.9765,0.0685,-0.0045
00250 DATA -0.064,0.912,0.168,-0.016,-0.0735,0.8155,0.2895,-0.0315
00260 DATA -0.072,0.696,0.424,-0.048,-0.0625,0.5625,0.5625
```

```
00270 DATA -0.0625,-0.048,0.424,0.696,-0.072,-0.0315,0.2895,0.8155
00280 DATA -0.0735,-0.016,0.168,0.912,-0.064,-0.0045,0.0685,0.9765
00290 DATA -0.0405,0,0,1,0
00300 FOR I=1 TO 4
00310 FOR J=1 TO 4
00320 E(I,J)=0
00330 NEXT J
00340 NEXT I
00350  :###  ####.#####      #####         #####.#####
00360 :#####     #####
00370 :####.####
00380 REM PRINT OPTION  IF A#1,ALL IS PRINTED
00390 A=1
00400 PRINT #5,  "-------------JUL--------------"
00410 PRINT #5
00420 PRINT #5, "------------------------------------"
00430 PRINT #5
00440 PRINT #5, "--------SAMPLE ITEMS------------"
00450 PRINT #5
00460 N=1
00470 IF END #1 THEN 00520
00480 INPUT #1,H(N)
00490 PRINT #5 USING 00370,H(N);
00500 N=N+1
00510 GO TO 00470
00520 N=N-1
```

```
00530 PRINT #5
00540 K=2.0
00550 S3=0
00560 S1=0
00570 FOR I=1 TO N
00580 S1=S1+H(I)
00590 S3=S3+H(I)*H(I)
00600 NEXT I
00610 H1=S1/N
00620 S5=SQR((S3-S1*S1/N)/(N-1))
00630 PRINT #5
00640 PRINT #5, "  SAMPLE AVERAGE IS ";H1
00650 PRINT #5, "  SAMPLE STANDARD DEVIATION IS ";S5
00660 PRINT #5
00670  PRINT #5, "---CLASS LIMITS---CLASSIFIED SAMPLE--CLASS PROBABILITI
00680 I=0
00690 V0=4-2.5*EXP(-.91629*H1/K/S5)
00700 V0=INT(1000*V0+0.5)/1000
00710 V2=V0/30
00720 FOR V1=V2 TO V0 STEP V2
00730 I=I+1
00740 IF V1>= 1.5 THEN 00770
00750 M(I)=K*S5*LOG(V1/1.5)/(-1.52715)+H1
00760 GO TO 00780
00770 M(I)=K*S5*LOG((4-V1)/2.5)/0.91629+H1
00780 NEXT V1
```

```
00790 FOR L=1 TO 31
00800 C(L)=0
00810 NEXT L
00820 I=1
00830 IF H(I)=0 THEN 00870
00840 K=1
00850 IF H(I)>M(K) THEN 00910
00860 GO TO 00890
00870 C(31)=C(31)+1
00880 GO TO 00920
00890 K=K+1
00900 GO TO 00850
00910 C(K)=C(K)+1
00920 IF I=N THEN 00950
00930 I=I+1
00940 GO TO 00830
00950 FOR I=1 TO 31
00960  D(I)=C(I)
00970 NEXT I
00980 A1=0
00990 FOR I=1 TO 31
01000 A1=A1+C(I)
01010 C(I)=A1/N
01020 PRINT #5 USING 00350,I,M(I),D(I),C(I)
01030 NEXT I
01040 I=1
```

```
01050 K=1
01060 L=1
01070 E(I,K)=E(I,K)+P(I,L)*P(K,L)
01080 IF L=30 THEN 01110
01090 L=L+1
01100 GOTO 01070
01110 IF K=4 THEN 01140
01120 K=K+1
01130 GOTO 01060
01140 IF I=4 THEN 01170
01150 I=I+1
01160 GOTO 01050
01170 I=1
01180 FOR K=1 TO 4
01190 E(K,I)=E(I,K)
01200 NEXT K
01210 PRINT #5
01220 Z=0.25
01230 PRINT #5, "--INITIAL Z VALUE= ";Z
01240 S2=Z
01250 C1=(((-8*Z+4)*Z+2)*Z-1)/16
01260 C2=(((24*Z-4)*Z-22)*Z+9)/16
01270 C3=(((-24*Z-4)*Z+22)*Z+9)/16
01280 C4=(((8*Z+4)*Z-2)*Z-1)/16
01290 IF A=1 THEN 01330
01300 PRINT #5
```

```
01310 PRINT #5, "CLASS 31 COEFS. ";C1;C2;C3;C4
01320 PRINT #5
01330 P(1,31)=0
01340 P(2,31)=C1+C4
01350 P(3,31)=C2-3*C4
01360 P(4,31)=C3+3*C4
01370 G(2,2)=E(2,2)+P(2,31)*P(2,31)
01380 G(2,3)=E(2,3)+P(2,31)*P(3,31)
01390 G(3,2)=G(2,3)
01400 G(2,4)=E(2,4)+P(2,31)*P(4,31)
01410 G(4,2)=G(2,4)
01420 G(3,3)=E(3,3)+P(3,31)*P(3,31)
01430 G(3,4)=E(3,4)+P(3,31)*P(4,31)
01440 G(4,3)=G(3,4)
01450 G(4,4)=E(4,4)+P(4,31)*P(4,31)
01460 G(1,1)=E(1,1)
01470 G(1,2)=E(1,2)
01480 G(1,3)=E(1,3)
01490 G(1,4)=E(1,4)
01500 G(2,1)=E(2,1)
01510 G(3,1)=E(3,1)
01520 G(4,1)=E(4,1)
01530 IF A=1 THEN 01620
01540 PRINT #5, "----G(I,J) MATRIX----"
01550 FOR I=1 TO 4
01560 PRINT #5, I
```

```
01570 FOR J=1 TO 4
01580 PRINT #5, G(I,J);
01590 NEXT J
01600 PRINT #5
01610 NEXT I
01620 FOR I=1 TO 4
01630 FOR J=1 TO 4
01640 Q(I,J)=0
01650 NEXT J
01660 NEXT I
01670 FOR I=1 TO 4
01680 Q(I,I)=1
01690 NEXT I
01700 J=1
01710 I=J
01720 IF G(I,J)<>0 THEN 01780
01730 IF I=4 THEN 01760
01740 I=I+1
01750 GOTO 01720
01760 PRINT #5, "SINGULAR MATRIX"
01770 GOTO 02420
01780 GOTO 01790
01790 K=1
01800 S=G(J,K)
01810 G(J,K)=G(I,K)
01820 G(I,K)=S
```

```
01830 S=Q(J,K)
01840 Q(J,K)=Q(I,K)
01850 Q(I,K)=S
01860 IF K=4 THEN 01890
01870 K=K+1
01880 GOTO 01800
01890 T=1/G(J,J)
01900 K=1
01910 G(J,K)=T*G(J,K)
01920 Q(J,K)=T*Q(J,K)
01930 IF K=4 THEN 01960
01940 K=K+1
01950 GOTO 01910
01960 L=1
01970 IF L=J THEN 02050
01980 T=-G(L,J)
01990 K=1
02000 G(L,K)=G(L,K)+T*G(J,K)
02010 Q(L,K)=Q(L,K)+T*Q(J,K)
02020 IF K=4 THEN 02050
02030 K=K+1
02040 GOTO 02000
02050 IF L=4 THEN 02080
02060 L=L+1
02070 GOTO 01970
02080 IF J=4 THEN 02110
```

```
02090 J=J+1
02100 GOTO 01710
02110 IF A=1 THEN 02210
02120 PRINT #5
02130 PRINT #5, "------INVERSE MATRIX--------"
02140 FOR I=1 TO 4
02150 PRINT #5, I
02160 FOR J=1 TO 4
02170 PRINT #5, Q(I,J);
02180 NEXT J
02190 PRINT #5
02200 NEXT I
02210 R(1)=0
02220 R(2)=0
02230 R(3)=0
02240 R(4)=0
02250 I=1
02260 R(1)=R(1)+P(1,I)*C(I)
02270 R(2)=R(2)+P(2,I)*C(I)
02280 R(3)=R(3)+P(3,I)*C(I)
02290 R(4)=R(4)+P(4,I)*C(I)
02300 IF I=31 THEN 02330
02310 I=I+1
02320 GOTO 02260
02330 I=1
02340 Y(I)=0
```

```
02350 J=1
02360 Y(I)=Y(I)+Q(I,J)*R(J)
02370 IF J=4 THEN 02400
02380 J=J+1
02390 GOTO 02360
02400 IF I=4 THEN 02430
02410 I=I+1
02420 GOTO 02340
02430 Y(5)=Y(2)-3*Y(3)+3*Y(4)
02440 IF A=1 THEN 02480
02450 PRINT #5, "---------SOLUTION NODES---------"
02460 PRINT #5, Y(1);Y(2);Y(3);Y(4);Y(5)
02470 PRINT #5
02480 Z1=-0.5
02490 Z2=0.5
02500 Z=(Z1+Z2)/2
02510 C1=(((-8*Z+4)*Z+2)*Z-1)/16
02520 C2=(((24*Z-4)*Z-22)*Z+9)/16
02530 C3=(((-24*Z-4)*Z+22)*Z+9)/16
02540 C4=(((8*Z+4)*Z-2)*Z-1)/16
02550 P1=C1*Y(2)+C2*Y(3)+C3*Y(4)+C4*Y(5)
02560 IF A=1 THEN 02580
02570 PRINT #5, "-----P1 NOW IS  ";P1
02580 IF ABS(P1-1)<0.001 THEN 02640
02590 IF P1>1 THEN 02620
02600 Z1=Z
```

```
02610 GOTO 02500
02620 Z2=Z
02630 GOTO 02500
02640 IF A=1 THEN 02670
02650 PRINT #5
02660 PRINT #5, "VALUE OF 100% INTERCEPT IS ";3.5+Z
02670 IF ABS(S2-Z)<0.001 THEN 02700
02680 S2=Z
02690 GOTO 01250
02700 PRINT #5
02710 PRINT #5, " CONVERGENCE STOP "
02720 PRINT #5
02730 PRINT #5,"-----SOLUTION NODES----"
02740 PRINT #5, Y(1);Y(2);Y(3);Y(4);Y(5)
02750 REM RELOCATE ORDINATES
02760 Y(7)=Y(5)
02770 Y(6)=Y(4)
02780 Y(5)=Y(3)
02790 Y(4)=Y(2)
02800 Y(3)=Y(1)
02810 Y(2)=0
02820 Y(1)=Y(3)
02830 FOR K=2 TO 5
02840 Z=-0.5
02850 A=(9*(Y(K)+Y(K+1))-Y(K-1)-Y(K+2))/16
02860 B=(11*(Y(K+1)-Y(K))+Y(K-1)-Y(K+2))/8
```

```
02870 C=(Y(K-1)-Y(K)-Y(K+1)+Y(K+2))/4
02880 D=(3*(Y(K)-Y(K+1))-Y(K-1)+Y(K+2))/2
02890 FOR J=1 TO 10
02900 Z=Z+0.1
02910 I1=(K-2)*10+J
02920 T(I1)=((D*Z+C)*Z+B)*Z+A
02930 NEXT J
02940 NEXT K
02950  PRINT #5
02960 PRINT #5, " INTERPOLATED VALUES "
02970 PRINT #5
02980 FOR I=1 TO 40
02990 PRINT #5, I;T(I),
03000 NEXT I
03010 PRINT #5
03020 STOP
03030 END
```

Notes For Program 2

Line #	Comment
3	Begins arrays at one.
17 - 29	Sliding Polynomial Coefficients for data set organized into 30 classes.
47 - 52	Input sample.
55 - 65	Compute average and standard deviation.
68 - 71	Set width of 30 uniform classes in v-scale.
72 - 78	Compute class boundaries in original data scale.
82 - 94	Tally the non-zero values of the sample into 30 classes. Zeroes tally into class 31.
98 - 103	Accumulate across classes and take ratios.
104 - 120	Compute the "Sums-of-Products" matrix for 30 classes of non-zeroes.
122 - 128	Initialize z of 31st class with trial value of 0.25 and compute Sliding Polynomial Coefficients of 31st class.
133 - 152	Augment the fixed "Sums-of-Products" matrix of 30 classes with Sums-of-Products of trial Coefficients of 31st class. Fifth node is assumed on extension of parabola through nodes 2, 3, & 4.
162 - 210	Invert the Sums-of-Products matrix.
221 - 232	Compute the Σ XY vector of least squares.
233 - 242	Multiply Σ XY vector by inverse matrix to get 4 nodes.
243	Extend parabola to fifth node.
248 - 255 258 - 263	Find new value of z where smoothing curve crosses P=100 line. Use reverse interpolation by interval-halving method.
267 - 269	If new z different from trial z of 100 percent crossing, repeat new trial solution.
276 - 282	After iteration to solution, Re-position the nodes and put in the boundary nodes.
283 - 294	Lay in the ensemble of 4 sliding polynomial arcs across 5 nodes.

Sample Output For Program 2

July

----------------------------------Sample Items----------------------------------

.1000	2.1300	2.6400	3.4500	.6900	.9900
.0300	.0000	1.6300	.7600	1.0400	.0000
.0000	7.8000	2.3600	1.3000	.6100	3.5100
4.1700	.9700	.3600	3.9100	.4100	8.9200
4.0100	.0000	1.5500	7.2400	.0000	.1500
.1500	.0000	.6400	.0300	3.4000	.9400
.5800	8.3100	2.0800	8.1800	1.9600	1.0200
1.0200	5.8900	1.4200	3.3500	4.9300	1.1700
1.4200	.0000	6.3800	.4800	.2300	11.2500
.3300	2.6200	1.2400	2.3100	11.8400	.4300
.9900	5.0800	.0000	5.1100	1.6500	9.4200
5.6900	.4600	.8600	1.8000	.8100	1.3200
1.4200	.2500	2.7200	4.1400	7.1400	8.0000
24.9400	.4100	3.5600	2.0600	.9900	4.2900
1.6800	.0000	2.2100	.0000	7.7200	2.2900
6.2500	2.1100	8.8400	2.1300	.0800	.5300
1.1200	.0500	.5100	.1000	1.8300	.5300
2.4100	5.3100	6.5800	5.8900	3.0500	1.5500
1.9300	.0500	.0800	2.5400	.0500	1.9600
.9400	.8100	.5800	.0800	.3000	15.3400
1.5000	3.7600	2.9700	5.3800	6.8300	.0000
.1800	1.7500	.9700	2.1100	.5600	3.1500
.0000	.9100	4.9500	8.5300	1.3000	2.1800
3.2500	.9900	.0000	.0000	.4300	.4300

Sample Average is 2.61806
Sample Standard Deviation is 3.40995

	Class Limits	Classified Sample	Class Probabilities
1	16.01421	1	.00694
2	12.91877	1	.01389
3	11.10806	2	.02778
4	9.82334	0	.02778
5	8.82683	3	.04861
6	8.01262	3	.06944
7	7.32422	3	.09028
8	6.72790	3	.11111
9	6.20191	3	.13194
10	5.73139	2	.14583
11	5.30576	3	.16667
12	4.91719	4	.19444
13	4.55973	0	.19444
14	4.22878	1	.20139
15	3.92068	3	.22222

Sample Output For Program 2 (continued)

	Class Limits	Classified Sample	Class Probabilities
16	3.63246	2	.23611
17	3.36173	4	.26389
18	3.10647	3	.28472
19	2.86502	2	.29861
20	2.63595	2	.31250
21	2.41066	2	.32639
22	2.17839	6	.36806
23	1.93865	8	.42361
24	1.69092	4	.45139
25	1.43466	6	.49306
26	1.16926	8	.54861
27	.89405	13	.63889
28	.60827	7	.68750
29	.31108	15	.79167
30	.00152	16	.90278
31	.00000	14	1.00000

Initial Z Value = .25

Convergence Stop

Solution Nodes

.154626	.322528	.899813	2.81714	6.07451

Interpolated Values

1	2.87814E-3	2	1.09206E-2	3	2.32395E-2	4	3.89469E-2	5	5.71549E-2
6	7.69755E-2	7	9.75208E-2	8	.117903	9	.137234	10	.154626
11	.169036	12	.180806	13	.191125	14	.20118	15	.212161
16	.225254	17	.24165	18	.262535	19	.289098	20	.322528
21	.357647	22	.390344	23	.423413	24	.459645	25	.501832
26	.552766	27	.615239	28	.692043	29	.785971	30	.899813
31	1.03124	32	1.17608	33	1.33431	34	1.50594	35	1.69097
36	1.8894	37	2.10124	38	2.32647	39	2.56511	40	2.81714

Note: Figure 3 is a plot of this sample output.

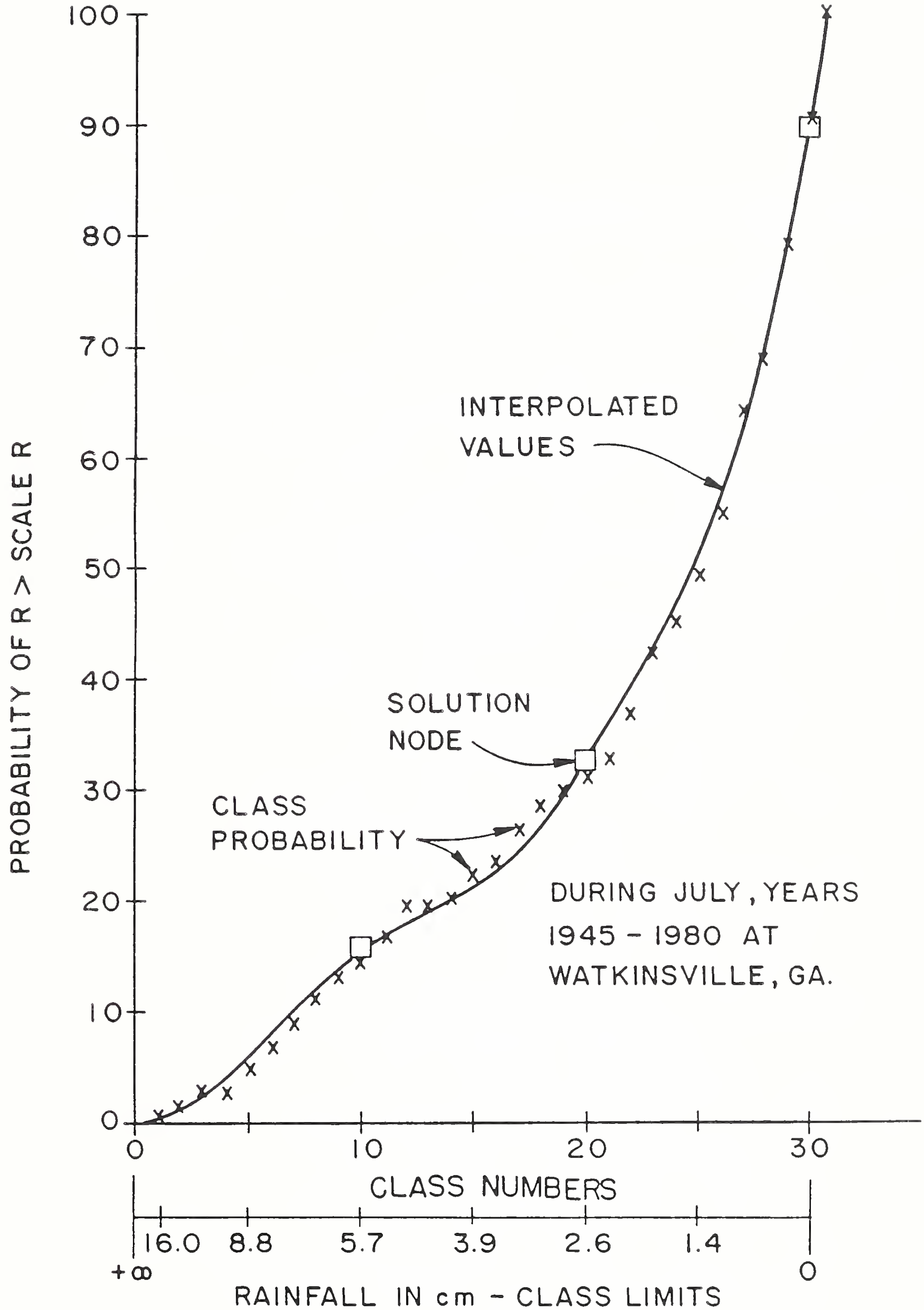


Figure 3. Distribution of Seven-Day Rainfall.

PROGRAM 3

Simulation of a number (NS) of synthetic samples of a number (NI) of items each.

Sample contains few zeroes.

Nodes were derived using Program 1.

```
C  SIMULATION WITH FEW ZEROES -- USE WITH PROGRAM NO. 1
      DIMENSION H(100),C(3,4),P(6),SORT(100,20),AT(80)
      READ(5,8080) (AT(I),I=1,80)
 8080 FORMAT(80A1)
      WRITE(6,8081) (AT(I),I=1,80)
 8081 FORMAT(' ',80A1)
      READ(5,1000) NR,N1,N2,NS,NI
 1000 FORMAT(3I5,2I10)
      WRITE(6,50) NR,N1,N2,NS,NI
   50 FORMAT(' ',5I8)
      READ(5,1001) (P(I),I=3,6)
 1001 FORMAT(4F10.6)
      WRITE(6,8181) (P(I),I=3,6)
 8181 FORMAT(' ',4F10.6)
      READ(5,1001) HB,SD,Q
      WRITE(6,8181) HB,SD,Q
      P(1)=P(3)
      P(2)=0.0
      CALL RAND(NR,N1,N2,R,1)
      DO 1002 I=1,3
      C(I,1)=(9*(P(I+1)+P(I+2))-P(I)-P(I+3))/16
      C(I,2)=(11*(P(I+2)-P(I+1))+P(I)-P(I+3))/8
      C(I,3)=(P(I)-P(I+1)-P(I+2)+P(I+3))/4
 1002 C(I,4)=(3*(P(I+1)-P(I+2))-P(I)+P(I+3))/2
      VS=(4.-2.5*EXP(-.91629*HB/Q/SD))/3.0
```

```
      DO 1003 M=1,NS
      DO 1004 I=1,NI
      CALL RAND(NR,N1,N2,R,2)
      IF (R.LT.P(3)) GO TO 1005
      IF (R.LT.P(4)) GO TO 1006
      GO TO 1007
1005  BL=0.0
      BR=VS
      IS=1
      GO TO 1008
1006  BL=VS
      BR=2.0*VS
      IS=2
      GO TO 1008
1007  BL=2.0*VS
      BR=3.0*VS
      IS=3
1008  A=(-BL)/(BR-BL)-0.5
      B=1/(BR-BL)
1100  Z=A+B*(BL+BR)/2.0
      PC=((C(IS,4)*Z+C(IS,3))*Z+C(IS,2))*Z+C(IS,1)
      IF(R.LT.PC) GO TO 1009
      BL=(Z-A)/B
      GO TO 1010
1009  BR=(Z-A)/B
1010  IF(ABS(BR-BL).LT.0.001) GO TO 1011
```

```
      GO TO 1100
 1011 V=(BL+BR)/2
      IF(V.LT.1.5) GO TO 1012
      H(I)=HB+Q*SD*ALOG((4.0-V)/2.5)/0.91629
      GO TO 1004
 1012 H(I)=HB-Q*SD*ALOG(V/1.5)/1.52715
 1004 CONTINUE
      DO 10 J=1,NI
      X6=H(J)
      DO 20 J1=1,NI
      IF(H(J1).GE.X6) GO TO 20
      H(J)=H(J1)
      H(J1)=X6
      X6=H(J)
   20 CONTINUE
   10 CONTINUE
      WRITE(6,1017) M
 1017 FORMAT(10X,'SAMPLE NO.',I4/)
      WRITE(6,1018) (K,H(K),K=1,NI)
 1018 FORMAT(' ',8(I4,F8.3))
      DO 2000 K=1,NI
      SORT(M,K)=H(K)
 2000 CONTINUE
 1003 CONTINUE
      DO 2010 K=1,NI
```

```
      DO 2005 I=1,NS
      CHECK=SORT(I,K)
      DO 2004 J=1,NS
      IF(SORT(J,K).GE.CHECK) GO TO 2004
      SORT(I,K)=SORT(J,K)
      SORT(J,K)=CHECK
      CHECK=SORT(I,K)
 2004 CONTINUE
 2005 CONTINUE
 2010 CONTINUE
      WRITE(6,2223) (L,L=1,NI)
 2223 FORMAT(' ',//11X,'  SORTED DATA ',/,3X,20(I2,4X))
      DO 2001 M=1,NS
      WRITE(6,2222) (SORT(M,K),K=1,NI)
 2222 FORMAT(' ',20F6.2)
 2001 CONTINUE
   70 STOP
      END
      SUBROUTINE RAND(NR,N1,N2,DRAW,IENT)
      DIMENSION TAB(10,10)
      IF(IENT.NE.1) GO TO 1003
      CALL RANDO(N1,N12,XRN)
      N1=N12
      II=INT(10.0*XRN)+1
      CALL RANDO(N2,N22,XRN)
      N2=N22
```

```
      JJ=INT(10.0*XRN)+1
      DO 1000 I=1,10
      DO 1001 J=1,10
      CALL RANDO(NR,NR2,XRN)
      NR=NR2
 1001 TAB(I,J)=XRN
 1000 CONTINUE
      IC=1
 1003 DRAW=TAB(II,JJ)
      CALL RANDO(NR,NR2,XRN)
      NR=NR2
      TAB(II,JJ)=XRN
      IF(MOD(IC,2).EQ.0) GO TO 1005
      CALL RANDO(N1,N12,XRN)
      N1=N12
      II=INT(10.0*XRN)+1
      GO TO 1006
 1005 CALL RANDO(N2,N22,XRN)
      N2=N22
      JJ=INT(10.0*XRN)+1
 1006 IC=IC+1
      RETURN
      END
      SUBROUTINE RANDO(IX,IY,YFL)
      IY=IX*65539
```

```
      IF(IY) 5,6,6
    5 IY=IY+2147483647+1
    6 YFL=FLOAT(IY)*0.4656613E-9
      RETURN
      END
****************************** DATA ************************************
SIMULATION WITHOUT ZEROES PERIOD 1
750367995392145       100        15
    .25925    .44244   1.00052   1.86506
  14.78448  11.89578       2.0
```

Notes For Program 3

Line #	Comment
19	Initialize the random number matrix.
20 - 24	Compute the sliding polynomial coefficients for 3 arcs.
25	Range of v for one arc.
28 - 31	Draw a random number and find which arc it falls in.
32 - 42	Initialize appropriate arc for interval-halving method.
43 - 58	Compute a value H(I) by reverse interpolation from random R, using interval-halving method.
59 - 67	Place sample items in order of magnitude.
67 - 71	Print ranked sample.
72 - 74	Store ranked sample in array.
76 - 86	Place each sample rank in order of magnitude across samples.
87 - 92	Print ranked sample ranks.
95 - 131	Random number generator. Random draw of a random number from a 10 x 10 matrix, and replacement.

Sample Output For Program 3

Simulation Without Zeros Period 1

75036	79953	92145	100	15
0.259250	0.442440	1.000520	1.865060	
14.784479	11.895780	2.000000		

Sample No. 1

1	46.718	2	30.680	3	29.356	4	26.183	5	23.859	6	20.557	7	13.336	8	9.787
9	7.806	10	7.367	11	5.007	12	4.751	13	4.427	14	2.269	15	0.573		

Sample No. 2

1	47.201	2	32.835	3	25.130	4	15.850	5	14.652	6	14.227	7	12.842	8	12.251
9	12.184	10	10.856	11	6.055	12	5.908	13	4.284	14	1.007	15	0.100		

Sample No. 3

1	43.154	2	33.664	3	31.759	4	31.400	5	29.445	6	20.812	7	13.813	8	8.180
9	7.886	10	7.004	11	5.611	12	5.611	13	3.874	14	3.848	15	3.459		

Sample No. 4

1	27.850	2	27.709	3	23.703	4	15.627	5	12.662	6	10.957	7	10.835	8	7.332
9	6.408	10	4.879	11	4.323	12	3.794	13	2.410	14	0.769	15	0.238		

Sample No. 5

1	41.084	2	31.451	4	30.999	4	30.179	5	26.293	6	24.368	7	11.469	8	9.882
9	9.305	10	7.680	11	7.110	12	3.821	13	3.674	14	2.156	15	1.841		

Sample Output For Program 3 (continued)

Sample No. 6 - Sample No. 95 Not Shown

Sample No. 96

1	31.198	2	26.111	3	17.581	4	16.816	5	15.748	6	13.703	7	11.824	8	11.219
9	8.572	10	7.576	11	7.297	12	5.822	13	4.492	14	2.788	15	2.634		

Sample No. 97

1	42.059	2	41.959	3	27.609	4	21.591	5	19.165	6	17.951	7	13.104	8	12.615
9	8.056	10	7.518	11	6.898	12	6.165	13	4.699	14	2.815	15	2.340		

Sample No. 98

1	31.123	2	29.401	3	16.915	4	13.521	5	12.691	6	11.018	7	9.078	8	7.321
9	6.480	10	5.311	11	4.453	12	3.620	13	2.070	14	1.653	15	1.022		

Sample No. 99

1	70.933	2	37.659	3	34.634	4	33.547	5	27.569	6	24.894	7	15.480	8	14.678
9	8.067	10	4.349	11	3.981	12	3.282	13	2.326	14	1.259	15	0.406		

Sample No. 100

1	42.728	2	39.683	3	38.842	4	37.774	5	35.835	6	29.717	7	21.469	8	13.447
9	10.621	10	10.404	11	5.574	12	5.235	13	3.214	14	2.481	15	1.624		

Sample Output For Program 3 (continued)

Sorted Data

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
72.63	51.44	50.03	45.80	36.00	30.80	24.66	23.78	20.67	15.78	12.46	11.63	11.57	8.24	6.47
70.93	50.29	48.75	42.16	35.84	29.72	23.92	23.07	20.27	13.98	12.20	10.25	9.00	6.10	3.70
60.22	50.29	42.06	40.48	34.42	27.27	23.61	21.48	14.97	13.44	12.13	9.35	8.89	6.01	3.51
57.83	49.30	41.47	38.52	31.30	27.00	22.23	17.99	14.87	12.25	10.86	9.33	7.38	5.55	3.50
56.37	48.75	40.39	37.77	29.60	26.00	21.73	16.41	14.52	11.73	10.60	9.15	7.04	4.99	3.46

Entry No. 6 - Entry No. 95 Not Shown

25.61	21.11	15.07	12.59	9.77	8.65	7.32	6.65	5.42	3.32	2.93	2.25	1.26	0.77	0.10
24.58	20.90	14.87	11.89	9.50	7.95	7.04	6.37	5.25	2.94	2.55	2.24	1.14	0.71	0.10
23.81	20.65	14.24	10.83	9.35	7.75	6.96	6.25	5.06	2.77	2.52	1.97	1.08	0.71	0.10
23.02	17.16	10.52	10.21	9.03	7.71	6.78	6.20	4.82	2.47	2.27	1.96	1.07	0.54	0.08
21.92	13.86	10.44	8.80	8.47	7.36	6.77	6.03	3.57	2.30	1.97	1.49	0.99	0.53	0.05

PROGRAM 4

Simulation of a number (NS) of synthetic samples of a number (NI) of items each.

Sample contains zeroes.

Nodes were derived using Program 2.

```
C  SIMULATION WITH ZEROES -- USE WITH PROGRAM NO. 2
      DIMENSION H(100),C(3,4),P(6),SORT(110,20),AT(80)
      READ(5,8080) (AT(I),I=1,80)
 8080 FORMAT(80A1)
      WRITE(6,8081) (AT(I),I=1,80)
 8081 FORMAT(' ',80A1)
      READ(5,1000) NR,N1,N2,NS,NI
 1000 FORMAT(3I5,2I10)
      WRITE(6,50) NR,N1,N2,NS,NI
   50 FORMAT(' ',6X/,5I8)
      READ(5,1001) (P(I),I=3,6)
 1001 FORMAT(4F10.6)
      WRITE(6,1001) (P(I),I=3,6)
      READ(5,1001) HB,SD,Q
      WRITE(6,1001) HB,SD,Q
      P(1)=P(3)
      P(2)=0.0
      CALL RAND(NR,N1,N2,R,1)
      DO 1002 I=1,3
      C(I,1)=(9*(P(I+1)+P(I+2))-P(I)-P(I+3))/16
      C(I,2)=(11*(P(I+2)-P(I+1))+P(I)-P(I+3))/8
      C(I,3)=(P(I)-P(I+1)-P(I+2)+P(I+3))/4
 1002 C(I,4)=(3*(P(I+1)-P(I+2))-P(I)+P(I+3))/2
      VS=(4.-2.5*EXP(-.91629*HB/Q/SD))/3.0
      DO 1003 M=1,NS
```

```
      DO 1004 I=1,NI
      CALL RAND(NR,N1,N2,R,2)
      IF (R.LT.P(3)) GO TO 1005
      IF (R.LT.P(4)) GO TO 1006
      IF (R.LT.P(5)) GO TO 1007
      H(I)=0.0
      GO TO 1004
 1005 BL=0.0
      BR=VS
      IS=1
      GO TO 1008
 1006 BL=VS
      BR=2.0*VS
      IS=2
      GO TO 1008
 1007 BL=2.0*VS
      BR=3.0*VS
      IS=3
 1008 A=(-BL)/(BR-BL)-0.5
      B=1/(BR-BL)
 1100 Z=A+B*(BL+BR)/2.0
      PC=((C(IS,4)*Z+C(IS,3))*Z+C(IS,2))*Z+C(IS,1)
      IF(R.LT.PC) GO TO 1009
      BL=(Z-A)/B
      GO TO 1010
 1009 BR=(Z-A)/B
```

```
1010 IF(ABS(BR-BL).LT.0.001) GO TO 1011
     GO TO 1100
1011 V=(BL+BR)/2
     IF(V.LT.1.5) GO TO 1012
     H(I)=HB+Q*SD*ALOG((4.0-V)/2.5)/0.91629
     GO TO 1004
1012 H(I)=HB-Q*SD*ALOG(V/1.5)/1.52715
1004 CONTINUE
     DO 10 J=1,NI
     X6=H(J)
     DO 20 J1=1,NI
     IF(H(J1).GE.X6) GO TO 20
     H(J)=H(J1)
     H(J1)=X6
     X6=H(J)
  20 CONTINUE
  10 CONTINUE
     WRITE(6,1017) M
1017 FORMAT(' '/,10X,'SAMPLE NO.',I4)
     WRITE(6,1018) (K,H(K),K=1,NI)
1018 FORMAT(' ',8(I4,F8.3)/)
     DO 2000 K=1,NI
     SORT(M,K)=H(K)
2000 CONTINUE
1003 CONTINUE
```

```
      DO 2010 K=1,NI
      DO 2005 I=1,NS
      CHECK=SORT(I,K)
      DO 2004 J=1,NS
      IF(SORT(J,K).GE.CHECK) GO TO 2004
      SORT(I,K)=SORT(J,K)
      SORT(J,K)=CHECK
      CHECK=SORT(I,K)
 2004 CONTINUE
 2005 CONTINUE
 2010 CONTINUE
      WRITE(6,2223) (L,L=1,NI)
 2223 FORMAT(' ',//11X,'  SORTED DATA '/,3X,20(I2,4X))
      DO 2001 M=1,NS
      WRITE(6,2222) (SORT(M,K),K=1,NI)
 2222 FORMAT(' ',20F6.2)
 2001 CONTINUE
   70 STOP
      END
      SUBROUTINE RAND(NR,N1,N2,DRAW,IENT)
      DIMENSION TAB(10,10)
      IF(IENT.NE.1) GO TO 1003
      CALL RANDO(N1,N12,XRN)
      N1=N12
      II=INT(10.0*XRN)+1
      CALL RANDO(N2,N22,XRN)
```

```
      N2=N22
      JJ=INT(10.0*XRN)+1
      DO 1000 I=1,10
      DO 1001 J=1,10
      CALL RANDO(NR,NR2,XRN)
      NR=NR2
 1001 TAB(I,J)=XRN
 1000 CONTINUE
      IC=1
 1003 DRAW=TAB(II,JJ)
      CALL RANDO(NR,NR2,XRN)
      NR=NR2
      TAB(II,JJ)=XRN
      IF(MOD(IC,2).EQ.0) GO TO 1005
      CALL RANDO(N1,N12,XRN)
      N1=N12
      II=INT(10.0*XRN)+1
      GO TO 1006
 1005 CALL RANDO(N2,N22,XRN)
      N2=N22
      JJ=INT(10.0*XRN)+1
 1006 IC=IC+1
      RETURN
      END
      SUBROUTINE RANDO(IX,IY,YFL)
```

```
      IY=IX*65539
      IF(IY) 5,6,6
    5 IY=IY+2147483647+1
    6 YFL=FLOAT(IY)*0.4656613E-9
      RETURN
      END
****************************** DATA **********************************
SIMULATION WITH ZEROES. WATKINSVILLE, GA. JULY
483389174482781       100        15
    .15463    .32253   0.89981   2.81714
   2.61806   3.40995       2.0
```

Notes For Program 4

Line #	Comment
18	Initialize the random number matrix.
19 - 23	Compute the Sliding Polynomial Coefficients.
24	Range of v for one arc.
27 - 30	Draw a random number and find which arc it falls in.
31 - 32	Zero if beyond P(5).
33 - 43	Initialize appropriate arc for interval-halving method.
44 - 59	Compute a value H(I) by reverse interpolation from random R, using interval-halving method.
60 - 68	Place sample in order of magnitude.
69 - 72	Print ranked sample.
73 - 75	Store ranked sample in array.
77 - 87	Place each sample rank in order of magnitude across samples.
88 - 92	Print ranked sample ranks.
96 -132	Random number generator. Random draw of a random number from a 10 x 10 matrix, and replacement.

Sample Output For Program 4

Simulation With Zeros. Watkinsville, GA. July

48338	91744	82781	100	15
0.154630	0.322530	0.899810	2.817140	
2.618059	3.409949	2.000000		

Sample No. 1

1	4.179	2	2.456	3	2.171	4	2.025	5	1.892	6	1.769	7	1.554	8	0.929
9	0.728	10	0.629	11	0.366	12	0.0	13	0.0	14	0.0	15	0.0		

Sample No. 2

1	10.024	2	7.608	3	6.623	4	6.012	5	2.293	6	1.973	7	0.697	8	0.694
9	0.494	10	0.206	11	0.197	12	0.127	13	0.029	14	0.0	15	0.0		

Sample No. 3

1	5.454	2	5.111	3	3.455	4	2.990	5	2.479	6	1.234	7	1.099	8	1.029
9	0.973	10	0.907	11	0.716	12	0.643	13	0.185	14	0.0	15	0.0		

Sample No. 4

1	16.665	2	6.320	3	3.354	4	2.707	5	2.656	6	1.591	7	1.518	8	1.420
9	1.412	10	1.234	11	1.213	12	1.181	13	0.595	14	0.289	15	0.039		

Sample No. 5

1	12.903	2	11.983	3	9.868	4	9.457	5	5.062	6	2.685	7	2.553	8	2.086
9	1.239	10	0.623	11	0.520	12	0.172	13	0.039	14	0.0	15	0.0		

Sample Output For Program 4 (continued)

Sample No. 6 - Sample No. 95 Not Shown

Sample No. 96

1	14.354	2	10.583	3	6.966	4	2.630	5	2.558	6	2.452	7	2.333	8	2.034
9	1.184	10	1.051	11	0.520	12	0.471	13	0.465	14	0.395	15	0.0		

Sample No. 97

1	6.618	2	6.021	3	4.173	4	2.738	5	2.660	6	2.057	7	1.648	8	1.526
9	1.488	10	1.368	11	1.338	12	1.278	13	0.926	14	0.0	15	0.0		

Sample No. 98

1	6.623	2	2.724	3	2.483	4	1.091	5	0.869	6	0.595	7	0.558	8	0.328
9	0.097	10	0.029	11	0.005	12	0.0	13	0.0	14	0.0	15	0.0		

Sample No. 99

1	8.189	2	7.420	3	3.026	4	2.890	5	2.181	6	1.722	7	1.648	8	1.458
9	1.381	10	1.381	11	1.107	12	0.580	13	0.334	14	0.289	15	0.0		

Sample No. 100

1	7.896	2	6.305	3	5.269	4	3.057	5	2.338	6	1.851	7	0.946	8	0.592
9	0.586	10	0.436	11	0.363	12	0.248	13	0.194	14	0.166	15	0.0		

Sample Output For Program 4 (continued)

Sorted Data

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
25.06	13.48	10.63	9.46	7.11	6.66	5.88	3.06	2.86	2.82	2.43	1.75	1.30	1.01	0.36
24.17	12.67	10.51	7.70	7.10	6.64	4.04	3.05	2.56	2.22	2.14	1.45	1.11	0.76	0.31
21.28	12.24	10.48	7.51	6.77	6.55	3.80	2.96	2.44	2.11	2.10	1.45	1.04	0.68	0.30
18.29	12.20	9.87	7.50	6.56	4.64	3.46	2.93	2.32	2.02	1.92	1.35	0.93	0.56	0.27
17.43	12.18	9.06	7.23	6.43	4.08	3.35	2.54	2.19	1.97	1.70	1.30	0.81	0.40	0.26

Entry No. 6 - Entry No. 95 Not Shown

5.09	2.72	2.17	1.62	1.33	1.02	0.71	0.51	0.35	0.13	0.08	0.0	0.0	0.0	0.0
4.18	2.60	1.94	1.56	1.27	0.78	0.70	0.42	0.32	0.11	0.04	0.0	0.0	0.0	0.0
3.17	2.47	1.93	1.41	0.96	0.74	0.57	0.37	0.16	0.06	0.02	0.0	0.0	0.0	0.0
3.12	2.46	1.86	1.30	0.87	0.59	0.56	0.35	0.10	0.03	0.00	0.0	0.0	0.0	0.0
2.90	2.24	1.64	1.09	0.85	0.58	0.35	0.33	0.04	0.02	0.0	0.0	0.0	0.0	0.0